A genetic algorithm used for solving one optimization problem

Cite as: AIP Conference Proceedings **1915**, 040052 (2017); https://doi.org/10.1063/1.5017400
Published Online: 12 December 2017

E. N. Shipacheva, A. A. Petunin, and I. M. Berezin

A Genetic Algorithm Used for Solving One Optimization Problem

E. N. Shipacheva[1, a)], A. A. Petunin[1, b)] and I. M. Berezin[1, 2, c)]

[1]*B. N. Yeltsin Ural Federal University, 19 Mira St., Ekaterinburg, 620002, Russia*
[2]*Institute of Engineering Science, Ural Branch of the Russian Academy of Sciences, 34 Komsomolskaya St., Ekaterinburg, 620049, Russia*

[a)]Corresponding author: e.n.shipacheva@urfu.ru
[b)]a.a.petunin@urfu.ru
[c)]berezin@imach.uran.ru

Abstract. A problem of minimizing the length of the blank run for a cutting tool during cutting of sheet materials into shaped blanks is discussed. This problem arises during the preparation of control programs for computerized numerical control (CNC) machines. A discrete model of the problem is analogous in setting to the generalized travelling salesman problem with limitations in the form of precursor conditions determined by the technological features of cutting. A certain variant of a genetic algorithm for solving this problem is described. The effect of the parameters of the developed algorithm on the solution result for the problem with limitations is investigated.

INTRODUCTION

Sheet cutting is a widespread technological process in industry. It is used for manufacturing blanks and finished articles of complex geometric shapes. The most technological and modern equipment for cutting a sheet material is CNC machines allowing the cutting process to be fully automated.

When describing a cutting route, one faces a problem of optimizing the route of a cutting tool [1]. If a standard "closed path" cutting technique (Fig. 1) is used, wherein the route has one tie-in point and one tie-off point and the cutting is without interruption, then the problem of optimization of the cutting tool route results in the problem of optimizing the blank run. Cutting time is used as the objective function.

In the general case, a continuous plurality of tie-in points may exist for each contour. In order to shift to the discrete model of the optimization task, we will assume that a finite number of tie-in and tie-off points are associated with each contour. A significant requirement of cutting is to maintain the tie-off order, according to which the cut-out of the contour is preceded by the cut-out of contours within it. This requirement is determined by the features of the cutting technology using CNC machines [1].

The discrete model of the problem is analogous in setting to the Generalized Traveling Salesman Problem (GTSP) with limitations in the form of precursor conditions determined by the technological features of cutting [2].

As is known, this problem, in terms of calculating difficulty, is related to the NP-complex class. In order to solve it, various discrete optimization algorithms are used. The genetic algorithm (GA) is one of the currently popular methods for solving optimization and modeling problems [3].

Mechanics, Resource and Diagnostics of Materials and Structures (MRDMS-2017)
AIP Conf. Proc. 1915, 040052-1–040052-4; https://doi.org/10.1063/1.5017400
Published by AIP Publishing. 978-0-7354-1607-9/$30.00

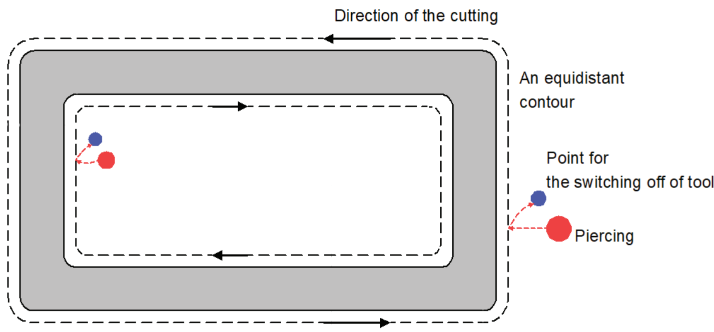


FIGURE 1. Scheme of the standard cutting technique

This study outlines a certain variant of the genetic algorithm for solving the previously mentioned traveling salesman problem with precursor conditions; the effect of the parameters of the developed algorithm on the length of the route for nesting is analyzed.

SETTING OF THE PROBLEM

Let us state the substantial setting out of the problem as follows: it is required to cut N external and internal closed contours in such a way that the total blank run of the cutting tool is minimal. As it has been recited above, a certain finite set of tie-in points is associated with each contour, and the tie-off point for the tool after cutting out the contour is unambiguously determined by the tie-in point selected.

The starting data for solving the task is the nesting obtained earlier. The nesting example is outlined in Fig. 2.

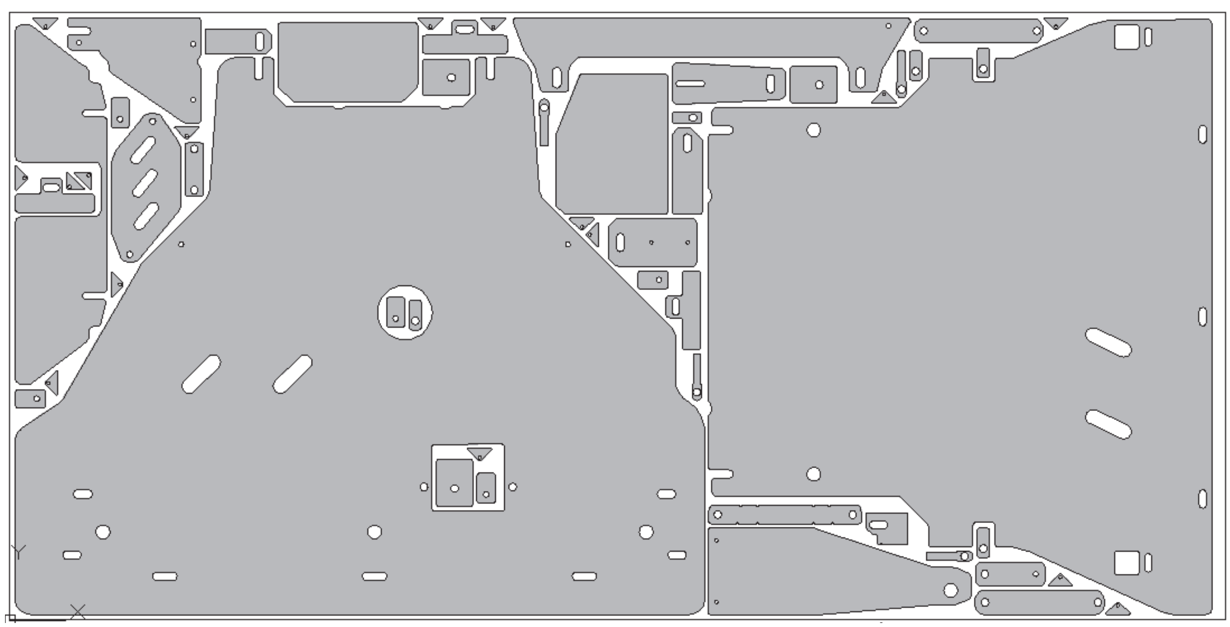


FIGURE 2. Nesting example developed in CAD T-Flex cut-out

Certain designations should be introduced. Let k^i be a number of possible tie-in points (and, correspondingly, the one for the tool tie-off points) in the contour i ($1 \leq i \leq N$). It is necessary to minimize the function like

$$L_{off}(R) = L\left(0, i_1^{s_1}\right) + \sum_{p=1}^{N-1} L\left(i_p^{s_p}, i_{p+1}^{s_{p+1}}\right) + L\left(i_N, 0\right), \quad (1)$$

wherein $L\ (i^s, j^t)$ is the distance between the tie-off point for the tool number s for the contour i and the tie-in point number t for the contour j ($s=1, \ldots, k^i$; $t=1, \ldots, k^j$). Herein R is a vector consisting of $2N$ elements: $R = (i_1, i_2, \ldots, i_N, s_1, s_2, \ldots, s_N)$, wherein s_m is the number of the selected tie-in point for the contour i_m ($1 \le s_m \le k_m$) [4].

The sequence $i_1, i_2, \ldots, i_N$ of the vector R is affected by the following limitations: if i_k is external, then all preceding internal contours should precede the contour i_k. These precursor conditions are determined, as noted above, by the technological features of sheet cutting, wherein the blank, after cutting out the contour limiting it, loses immobility and does not allow for correct positioning of the tools for cutting out the internal contours. We will use the Euclidean distance as a metric on the plurality of point pairs on the surface in the task examined.

DESCRIPTION OF THE ALGORITHM

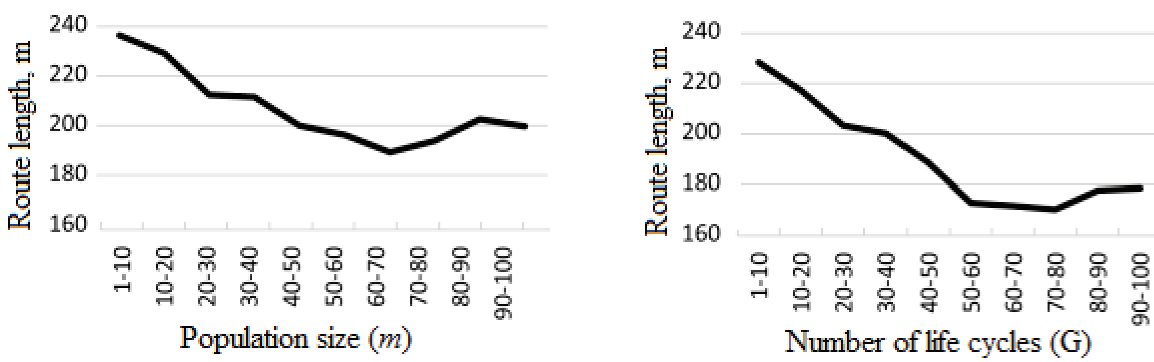
The general scheme of the implemented genetic algorithm is as follows:
Forming the starting population.
Selection of parents from the population.
Crossing the parents (reproduction operator).
Mutation of the subjects of the current population.
Forming the new population (selection operator).
Checking the achievement of the set number of life cycles.
Transition to the step 2.

We will introduce some basic terms: an *individual (subject)* is a certain solution of the problem (route – vector R); a *chromosome* is a pair (cluster number); the sequence of chromosomes determines the route. For the individual according to the formula (1), the objective function is calculated, which is a measure of the quality of the solution in the search space.

The parameters of the implemented genetic algorithm are as follows: m is the power of the population, i.e. the number of individuals comprised in it; m_r is the percentage of population subjects participating in the crossing; m_d is the percentage of population subjects subjected to the mutation operator; G is the number of generations (number of algorithm generations).

ALGORITHM IMPLEMENTATION

In order to realize the algorithm, we have developed a program using the Microsoft Visual Studio 2015 software and C# programming language, implementing the genetic algorithm. The investigations were carried out for the nesting depicted in Fig. 2. The nesting example obtained in the Russian CAD tool T-Flex [5] Cut-Out contains 56 parts (155 internal and external contours).



(a) (b)

FIGURE 3. Plot of the route length vs. the population size (a) and the number of life cycles (b)

Figure 3a shows a plot of the route length vs. population size (m). The parameters set for the algorithm are $G = 70$, $m_r = 20\%$, $m_d = 10\%$. One can see from the plot that an increase in the population size leads to a decrease in the value of the objective function.

Figure 3b shows a plot of the route length vs. the number of life cycles set by the user (number of iterations G). The parameters set for the algorithm are m = 100, m_r = 20%, m_d = 10%. An increase in the number of iterations leads to a decrease in the route length. This property follows from the ability of the algorithm to improve the population quality from generation to generation.

CONCLUSION

During this investigation, a genetic algorithm of solving the generalized traveling salesman problem with limitations in the form of precursor conditions, to which the problem of optimizing the blank run for the cutting tool during cut-out of sheet materials using CNC machines can be reduced, has been implemented. Further improvement of the algorithm consists in solving the following problems: elimination of the local extremum, alteration of the "crossing" operator, increasing the effect of the "mutation" operator, combining this algorithm with other optimization algorithms at the step of forming the initial population, taking into account various technological cutting requirements, such as the "blank hardness" and "material hardness" rule [4].

ACKNOWLEDGMENTS

The work was supported by the Russian Foundation for Basic Research, grant No. 17-08-01385.

REFERENCES

1. P. A. Chentsov and A. A. Petunin, IFAC-PapersOnLine, *Management and Control MIM,* Proceedings of the 8th IFAC Conference on Manufacturing Modelling, Troyes, 2016, edited by Alexandre Dolgui *et al*. (2016), Vol. 49–12, p. 645; A. G. Chentsov, P. A. Chentsov, A. A. Petunin and A. N. Sesekin, *ibid*. Vol. 49–12, p. 640.
2. R. Dewil, P. Vansteenwegen, D. Cattrysse, M. Laguna, and T. Vossen, International Journal of Production Research **53** (6), 1761 (2015).
3. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley Professional, 1989), p. 432.
4. A. A. Petunin, AIP Conference Proceedings, *41st International Conference on Applications of Mathematics in Engineering and Economics*, Sozopol, 2015, edited by Vesela Pasheva *et al*. (2015), Vol. 1690, p. 060002-1.
5. T-Flex CAD- top systems, available at http://www.tflex.com